REMARKS

I. Introduction

In response to the Office Action dated November 25, 2002, claims 1, 22, and 43 have been amended. Re-examination and re-consideration of the application, as amended, is requested.

II. Allowable Subject Matter

Claims 64-105 are allowed as they are patently distinct over the art of record. Applicants appreciate the indication of allowance. However, Applicants traverse the rejection of the claims as indicated below.

III. Prior Art Rejections

In pages (2)-(3) of the Office Action, claims 1-63 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kiernan et al., U.S. Patent No. 5,701,137 (Kiernan) in view of Weidenfeller et al., U.S. Patent No. 6,028,602 (Weidenfeller).

Specifically, claim 1 was rejected as follows:

> As per claim 1, Kiernan discloses a system for displaying a tree structure for representing hierarchical data in programmed computer comprising:
>
> selecting one or more objects on the original tree to be contained in the customized tree in response to user input (col. 6, lines 26-61, col. 7, lines 10-18);
>
> linking the selected objects in a user-specified manner (fig. 8b, col. 3, lines 1-8, col. 7, lines 20-48).
>
> Kiernan does not explicitly teach wherein the one or more objects are located in disparate places across different branches of the original tree. However, Weindenfeller discloses that one or more objects are located in disparate places across different branches of the original tree (fig. 7, col. 3, lines 17-29, col. 5, lines 14-25, 66-67, col. 6, lines 1-19, col. 9, lines 1-15). Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Keiman with Weidenfeller to implement the step of one or more objects are located in disparate places across different branches of the original tree in order to provide users the capability to perform operations on an object tree in a graphical user interface to display the user's selected object within the content of its hierarchy and also display related objects defined by user specified criteria. The selection's object makes it much easier to ensure that all and only the elements which should be selected are selected.

In reply to the prior response, the Office Action provides:

> Applicant's arguments filed on 10/16/02 have been fully considered but they are not persuasive.
>
> Applicants argue that the linking of any object together in a user-definable manner is not mentioned or suggested anywhere in Kiernan ('137).
>
> In response, the examiner respectfully disagrees. Kiernan et al ('137) does disclose linking the selected objects in a user-specified manner at column 3, lines 2-8 that "in a fist window, the

> application displays a master tree control corresponding to a portion of the tree structure. In response to a user command to separate the tree into another tree, the application creates a new tree control starting from the node selected by the user. Displayed in a new window, **the newly created tree control is a client of the master tree. The master tree control manages the underlying data representing a tree structure, and it communicates with one or more clients to update their display status."** And at column 7, lines 29-48 that "When new instances are created during the ripping process, the instances have a **master-client relationship. A master tree control owns the data associated with a tree.** A client tree control includes display status data, but must refer back to the master for data regarding nodes associated with it. This master-client relationship allows the application to process the underlying tree data in an organized fashion because there is only one copy of it, and all changes must occur through this "master" copy... When a user changes a node in a tree control, the master updates the underlying data and sends messages to the clients to update their display status. When the application creates a tree control, it specifies a master or client style flag to specify whether the new control is a master or a client. The application **binds the underlying node data and display data with the master control."** From the above passages, it is clear that Kiernan does indeed disclose the claimed "linking the selected objects in a user-specified manner".

Applicants traverse the above rejections. Specifically, neither Kiernan nor Wiedenfeller teach, disclose or suggest objects that are selected from disparate places across different branches of a tree and linked together to each other in the new tree in a user-selectable manner.

Rejected independent claims 1, 22, and 43, are generally directed to objects selected from disparate places in an original tree and linking those objects together (to each other) in the customizable tree in a user-specified manner. In other words, the selected objects are linked to one another as specified by a user in a new tree.

The cited references do not teach nor suggest these various elements of Applicants' independent claims.

Kiernan merely describes a method for interactive display of a graphical tree structure in a windowing environment. A tree control graphically represents hierarchical data. The user can separate a portion of a tree control at a node and create a new tree control for viewing and editing. Changes to a newly created tree control propagate through to related tree controls. As described in the preliminary amendment and as clearly set forth throughout Kiernan, Kiernan merely separates a portion of one tree into another tree:

> ...The user can separate a portion of a tree control at a node and create a new tree control for viewing and editing. Changes to a newly created tree control propagate through to the related tree controls. (See Abstract)
>
> ...In one implementation of the invention, the user can separate or "rip" a graphical tree structure into a smaller, more manageable tree for viewing and editing. (See col. 2, lines 53-56)
>
> ...In response to a user command to separate the tree into another tree, the application creates a new tree control starting from the node selected by the user. (See col. 3, lines 4-6).

Thus, unlike the present invention, in Kiernan, a node or particular location within a tree is identified and selected. Thereafter, the node and the nodes in the tree hierarchically below that node are "separated" and created in a new window. The mere separation of a node from a tree into a new window for editing is clearly different and distinguishable from a fully customizable tree where objects are selected from disparate places across different branches of a tree and linked together in the new tree in a user-selectable manner. In this regard, Kiernan fails to even remotely suggest such the fully customizable tree as claimed. Further, the ability to link together various objects from disparate locations to each other in a user-definable manner is not even remotely suggested.

The Office Action submits that having a master control and an underlying node teaches linking objects in a user-specified manner. However, as described above, Kiernan's client node is merely a subset of the master node (i.e., the client is a ripped tree that starts from the node selected by the user). Thus, the client node is a single tree or branch from the master tree. In this regard, Kiernan's client node does not comprise objects from disparate places across different branches of the master node. The claims provide for customizing and linking objects from different branches. Further, the claims provide that the objects are linked to each other. Kiernan fails to teach these claimed element. In this regard, linking data in a client tree to a master tree control that manages the data (i.e., the underlying node data and display data are bound to the master control) is clearly different from linking nodes within the client tree together (to each other) when those nodes in the client tree are from disparate locations in the master tree. Binding data in a master-client relationship is not even remotely equivalent to linking objects to each other in a single customizable tree.

Further, the Office Action admits that "Kiernan does not explicitly teach wherein one or more objects are located in disparate places across different branches of the original tree." Instead, the Office Action relies on Weidenfeller to teach this element. However, Weidenfeller also fails to teach, disclose, or suggest linking the objects selected from one tree in a new tree to each other in a user-customizable manner.

The present invention as claimed provides for two distinct elements – (1) objects from multiple disparate branches are selected; and (2) the selected objects (which are from the disparate branches) are linked to each other in a user-specified manner. These two combined features enable

a tree to be fully customizable and provides the user with great flexibility. Neither Kiernan nor Wiedenfeller provide such combined features. Instead, as described above, Kiernan provides for ripping a branch and creating a new tree – objects from disparate branches cannot be selected. Further, Wiedenfeller merely provides for placing objects into a flat bucket. In this regard, Weidenfeller discloses the use of a "bucket" that contains a subset of data from a tree (see col. 5, lines 2-8). The user selects items from a tree browser to go into the bucket (see col. 5, lines 47-65). However, Weidenfeller fails to suggest linking objects (to each other) in the buckets or in a new customized tree that contains objects from the original tree.

Instead, Weidenfeller suggests that selections in a bucket window can be added to the original tree browser window (see col. 8, lines 11-13). In other words, selections in the bucket windows can be copied back to one or more multiple tree browser windows (see col. 8, lines 31-34). However, Weidenfeller contains no teaching nor suggestion for linking the objects selected in the first window into a second customized "tree" in a user specified manner. In other words, while the claims provide for linking the objects (from disparate places) from a tree into a customized tree window, Weidenfeller merely suggests placing items into a bucket. There is no linking of the disparately located objects in the bucket in a particular user-specified manner. Instead, the objects are just placed into the bucket. Further, while Weidenfeller may teach that a bucket window can be viewed as a tree list (see col. 9, lines 1-10), it still fails to teach the linking of selected objects in the tree list to each other in a user-customized manner. Merely placing items into a bucket and viewing the bucket as a tree list is not equivalent and does not render obvious customizing the links of the objects in the tree in a user-specified manner as claimed.

Thus, neither separately or in combination, do Kiernan or Weidenfeller provide the ability to create a fully customizable tree that consists of objects retrieved from disparate locations in an original tree. Instead, Kiernan and Weidenfeller each independently teach elements that are distinguishable from the claimed invention. As described above, Kiernan merely creates a tree that is a subset of an original tree and links the original tree and subset tree in a relationship to each other. The present invention provides linking the selected objects to each other in the customized tree in a user customized manner. The claims do not provide for linking the selected objects in the customized tree with the objects in the original tree (as described by Kiernan). In this regard, there are nonobvious differences between linking objects to each other in a new tree and creating a

master-client relationship where data for an object in one tree is bound to the data for the object in a master tree.

Even when combined, the references teach away from Applicants' invention. For example, the combined references would teach ripping a portion of a tree from Kiernan into a bucket of Wiedenfeller. Such a teaching fails to provide for taking items from disparate locations, placing them into a new tree, and linking the items in the new tree to each other in a user-specified manner.

Moreover, the various elements of Applicants' claimed invention together provide operational advantages over Kiernan and Weidenfeller. In addition, Applicants' invention solves problems not recognized by Kiernan and Weidenfeller.

Thus, Applicants submit that independent claims 1, 22, 43, and 63 are allowable over Kiernan and Weidenfeller. Further, dependent claims 2-21, 23-42, and 44-63 are submitted to be allowable over Kiernan and Weidenfeller in the same manner, because they are dependent on independent claims 1, 22, 43, and 64, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-21, 23-42, and 44-63 recite additional novel elements not shown by Kiernan and Weidenfeller.

IV. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: February 24, 2003

By: _____
Name: George H. Gates
Reg. No.: 33,500

GHG/JSF/sjm/amb

G&C 30571.285-US-U1